

Amendment to the Claims

Kindly cancel claims 1-22, 24 and 27-44, without prejudice, and add claims 45-64, as set forth below. In compliance with the Revised Amendment Format published in the Official Gazette on February 25, 2003, a complete listing of claims is provided herein. The changes in the amended claims are shown by strikethrough (for deleted matter) and underlining (for added matter), with the exception that double brackets are used to indicate deleted matter if strikethrough cannot be easily perceived.

1. (Canceled)
2. (Canceled)
3. (Canceled)
4. (Canceled)
5. (Canceled)
6. (Canceled)
7. (Canceled)
8. (Canceled)
9. (Canceled)
10. (Canceled)
11. (Canceled)
12. (Canceled)
13. (Canceled)
14. (Canceled)
15. (Canceled)

16. (Canceled)
17. (Canceled)
18. (Canceled)
19. (Canceled)
20. (Canceled)
21. (Canceled)
22. (Canceled)
23. (Previously canceled)
24. (Canceled)
25. (Previously canceled)
26. (Previously canceled)
27. (Canceled)
28. (Canceled)
29. (Canceled)
30. (Canceled)
31. (Canceled)
32. (Canceled)
33. (Canceled)
34. (Canceled)
35. (Canceled)

36. (Canceled)
37. (Canceled)
38. (Canceled)
39. (Canceled)
40. (Canceled)
41. (Canceled)
42. (Canceled)
43. (Canceled)
44. (Canceled)
45. (New) A method of restoring debugging breakpoints, said method comprising:
 - having a breakpoint that is set to a selected step of a first version of source code of a program, said program being absent embedded debug commands;
 - creating an instruction profile for the selected step, said instruction profile comprising one or more attributes of one or more machine instructions generated for the selected step and one or more attributes of zero or more other machine instructions generated for the first version of source code; and
 - automatically restoring the breakpoint to the selected step of a modified program, in response to modification of the first version of source code to provide the modified program having a second version of source code, wherein the selected step is at a different location within the modified program, and wherein the automatically restoring comprises comparing one or more attributes of one or more machine instructions generated for the second version of source code with one or more attributes of the instruction profile created based on the first version of source code to determine the different location.

46. (New) The method of claim 45, wherein the comparing comprises comparing one or more operation codes of the one or more machine instructions generated for the second version of source code with one or more operation codes of the instruction profile to determine which machine instruction of the modified program corresponds most closely to the selected step.

47. (New) The method of claim 45, wherein the instruction profile further comprises a source line number for the selected step and a length of the first version of source code, and wherein the automatically restoring comprises using the source line number and length to determine a starting point within the modified program to select the one or more machine instructions generated for the second version to be used in the comparing.

48. (New) The method of claim 45, wherein the comparing yields one or more difference counts and a difference count having a smallest value indicates the different location.

49. (New) The method of claim 45, wherein the different location is identified by a substantial match between one or more attributes of the instruction profile and one or more attributes of one or more machine instructions of the modified program.

50. (New) The method of claim 45, wherein the creating comprises choosing a number of machine instructions to be included in the instruction profile.

51. (New) The method of claim 50, wherein the choosing comprises:

selecting a number of instructions to be included in a calibration profile;

generating the calibration profile for a chosen line of the program, said calibration profile having the selected number of instructions for said chosen line;

comparing one or more attributes of said calibration profile to one or more attributes of at least one line of code of the program to obtain a result;

determining whether the result is an unambiguous result; and

repeating, zero or more times, said selecting, said generating, said comparing, and said determining until the determining indicates an unambiguous result, wherein the selected number of instructions increases at each iteration, and wherein the selected number of instructions indicates, when there is an indication of an unambiguous result, the number of machine instructions to be included in the instruction profile.

52. (New) The method of claim 45, wherein said automatically restoring is performed by a debugger.

53. (New) A system of restoring debugging breakpoints, said system comprising:

a breakpoint that is set to a selected step of a first version of source code of a program, said program being absent embedded debug commands;

an instruction profile created for the selected step, said instruction profile comprising one or more attributes of one or more machine instructions generated for the selected step and one or more attributes of zero or more other machine instructions generated for the first version of source code;

a modified program having a second version of source code; and

means for automatically restoring the breakpoint to the selected step of the modified program, in response to modification of the first version of source code to provide the modified program, wherein the selected step is at a different location within the modified program, and wherein the means for automatically restoring comprises means for comparing one or more attributes of one or more machine instructions generated for the second version of source code with one or more attributes of the instruction profile created based on the first version of source code to determine the different location.

54. (New) The system of claim 53, wherein the means for comparing comprises means for comparing one or more operation codes of the one or more machine instructions generated for the second version of source code with one or more operation codes of the

instruction profile to determine which machine instruction of the modified program corresponds most closely to the selected step.

55. (New) The system of claim 53, wherein the instruction profile further comprises a source line number for the selected step and a length of the first version of source code, and wherein the means for automatically restoring comprises means for using the source line number and length to determine a starting point within the modified program to select the one or more machine instructions generated for the second version to be used in the comparing.

56. (New) The system of claim 53, wherein the comparing yields one or more difference counts and a difference count having a smallest value indicates the different location.

57. (New) The system of claim 53, wherein the different location is identified by a substantial match between one or more attributes of the instruction profile and one or more attributes of one or more machine instructions of the modified program.

58. (New) The system of claim 53, further comprising means for choosing a number of machine instructions to be included in the instruction profile, wherein the means for choosing comprises:

means for selecting a number of instructions to be included in a calibration profile;

means for generating the calibration profile for a chosen line of the program, said calibration profile having the selected number of instructions for said chosen line;

means for comparing one or more attributes of said calibration profile to one or more attributes of at least one line of code of the program to obtain a result;

means for determining whether the result is an unambiguous result; and

means for repeating, zero or more times, said selecting, said generating, said comparing, and said determining until the determining indicates an unambiguous result, wherein the selected number of instructions increases at each iteration, and wherein the selected number of instructions indicates, when there is an indication of an unambiguous result, the number of machine instructions to be included in the instruction profile.

59. (New) An article of manufacture comprising:

at least one computer usable medium having computer readable program code logic to restore debugging breakpoints, the computer readable program code logic comprising:

logic in which a breakpoint is set to a selected step of a first version of source code of a program, said program being absent embedded debug commands;

create logic to create an instruction profile for the selected step, said instruction profile comprising one or more attributes of one or more machine instructions generated for the selected step and one or more attributes of zero or more other machine instructions generated for the first version of source code; and

restore logic to automatically restore the breakpoint to the selected step of a modified program, in response to modification of the first version of source code to provide the modified program having a second version of source code, wherein the selected step is at a different location within the modified program, and wherein the restore logic comprises compare logic to compare one or more attributes of one or more machine instructions generated for the second version of source code with one or more attributes of the instruction profile created based on the first version of source code to determine the different location.

60. (New) The article of manufacture of claim 59, wherein the compare logic comprises logic to compare one or more operation codes of the one or more machine instructions generated for the second version of source code with one or more operation codes of the

instruction profile to determine which machine instruction of the modified program corresponds most closely to the selected step.

61. (New) The article of manufacture of claim 59, wherein the instruction profile further comprises a source line number for the selected step and a length of the first version of source code, and wherein the restore logic comprises use logic to use the source line number and length to determine a starting point within the modified program to select the one or more machine instructions generated for the second version to be used in the comparing.

62. (New) The article of manufacture of claim 59, wherein the comparing yields one or more difference counts and a difference count having a smallest value indicates the different location.

63. (New) The article of manufacture of claim 59, wherein the different location is identified by a substantial match between one or more attributes of the instruction profile and one or more attributes of one or more machine instructions of the modified program.

64. (New) The article of manufacture of claim 59, wherein the logic to create comprises choose logic to choose a number of machine instructions to be included in the instruction profile, wherein the choose logic comprises:

select logic to select a number of instructions to be included in a calibration profile;

generate logic to generate the calibration profile for a chosen line of the program, said calibration profile having the selected number of instructions for said chosen line;

compare logic to compare one or more attributes of said calibration profile to one or more attributes of at least one line of code of the program to obtain a result;

determine logic to determine whether the result is an unambiguous result; and

repeat logic to repeat, zero or more times, said selecting, said generating, said comparing, and said determining until the determining indicates an unambiguous result, wherein the selected number of instructions increases at each iteration, and wherein the selected number of instructions indicates, when there is an indication of an unambiguous result, the number of machine instructions to be included in the instruction profile.